Package 'hlaR'

October 24, 2022

Type Package

Title Tools for HLA Data

Version 0.1.5

Description A streamlined tool for eplet analysis of donor and recipient HLA (human leukocyte antigen) mismatch. Messy, low-resolution HLA typing data is cleaned, and imputed to high-resolution using the NMDP (National Marrow Donor Program) haplotype reference database <https://haplostats.org/haplostats>. High resolution data is analyzed for overall or single antigen eplet mismatch using a reference table (currently supporting 'HLAMatchMaker' <http://www.epitopes.net> versions 2 and 3). Data can enter or exit the workflow at different points depending on the user's aims and initial data quality.

Depends R (>= 4.1.0)

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.1

Suggests knitr, rmarkdown

URL https://pubmed.ncbi.nlm.nih.gov/35101308/,
https://emory-larsenlab.shinyapps.io/hlar_shiny/

BugReports https://github.com/LarsenLab/hlaR/issues

VignetteBuilder knitr

Imports devtools, tidyverse, dplyr, reshape2, schoolmath, tibble, stringr, purrr, tidyr, utils, readr, janitor

NeedsCompilation no

Author Joan Zhang [aut, cre],
Aileen Johnson [aut],
Christian P Larsen [cph, aut]

Maintainer Joan Zhang <joan.zhang@emory.edu>

Repository CRAN

Date/Publication 2022-10-24 21:35:04 UTC

R topics documented:

`CalAlleleMismFreq` *Evaluate the frequency of specific allele mismatches*

Description

This function evaluates allele level mismatch between donor and recipient and then presents the most commonly mismatched alleles. This function is most effectively used to study the most common mismatches within a transplant population.

Usage

```
CalAlleleMismFreq(dat_in, nms_don = c(), nms_rcpt = c())
```

Arguments

`dat_in`	A data frame of clean HLA typing data.
`nms_don`	A vector of column names of donor's alleles, must be length of 2.
`nms_rcpt`	A vector of column names of recipient's alleles, must be length of 2.

Value

A data frame of donor's mismatched alleles with frequency > 1. No mismatch is calculated if input alleles are NA.

Examples

```
dat <- read.csv(system.file("extdata/example", "HLA_MisMatch_test.csv", package = "hlaR"))
don <- c("donor.a1", "donor.a2")
rcpt <- c("recipient.a1", "recipient.a2")
re <- CalAlleleMismFreq(dat_in = dat, nms_don = don, nms_rcpt = rcpt)
```

CalAlleleTopN *topN most frequent HLA alleles*

Description

Input cleaned HLA(Human Leukocyte Antigen) data for a population of transplant donors and recipients to determine the most common alleles represented in the population.

Usage

```
CalAlleleTopN(dat_in, nms_don = c(), nms_rcpt = c(), top_n = 5)
```

Arguments

dat_in	A data frame with clean HLA typing data.
nms_don	A vector of donor's allele name(s).
nms_rcpt	A vector of recipient's allele name(s).
top_n	Number of alleles to return. Default is 5.

Value

A tibble of top_n most frequent alleles.

Examples

```
dat <- read.csv(system.file("extdata/example", "HLA_MisMatch_test.csv", package = "hlaR"))
don <- c("donor.a1", "donor.a2")
rcpt <- c("recipient.a1", "recipient.a2")
re <- CalAlleleTopN(dat_in = dat, nms_don = don, nms_rcpt = rcpt, top_n = 2)
```

CalEpletMHCI *Calculate class I HLA eplet mismatch*

Description

Use high resolution HLA(Human Leukocyte Antigen) class I data to calculate class I eplet mismatch for a population of donors and recipients. Mismatch is calculated using logic from 'HLAMatchMaker', developed by Rene Dusquesnoy. Current reference tables supported are 'HLAMatchMaker' v2 and v3.

Usage

```
CalEpletMHCI(dat_in, ver = 2)
```

Arguments

dat_in	A dataframe of recipient and donor's high resolution MHC I data. Each recipient and donor pair are linked by are the "pair_id" column and differentiated by the "subject_type" column.
ver	Version number of HLAMatchMaker based eplet reference table to use.

Value

A list of data tables. - 'single_detail': single molecule class I MHC eplet mismatch table, including mismatched eplet names and the count of eplets mismatched at each allele. - 'overall_count': original input data appended with total count of mismatched eplets.

Examples

```
dat<-read.csv(system.file("extdata/example","MHC_I_test.csv",package="hlaR"),sep=",",header=TRUE)
re <- CalEpletMHCI(dat_in = dat, ver = 3)
```

CalEpletMHCII *Calculate class II HLA eplet mismatch.*

Description

Use high resolution HLA(Human Leukocyte Antigen) class II data to calculate class II eplet mismatch for a population of donors and recipients. Mismatch is calculated using logic from 'HLAMatchMaker', developed by Rene Dusquesnoy. Current reference tables supported are 'HLAMatchMaker' v2 and v3. Note: interlocus info only available in v3 reference tables.

Usage

```
CalEpletMHCII(dat_in, ver = 2)
```

Arguments

dat_in	A dataframe of recipient and donor's high resolution MHC II data. Each recipient and donor pair are linked by are the "pair_id" column and differentiated by the "subject_type" column.
ver	Version number of HLAMatchMaker based eplet reference table to use.

Value

A list of data tables. - 'single_detail': single molecule class II MHC eplet mismatch table, including mismatched eplet names and the count of eplets mismatched at each allele. - 'overall_count': original input data appended with total count of mismatched eplets.

Examples

```
dat <- read.csv(system.file("extdata/example","MHC_II_test.csv",package="hlaR"),sep=",",header=TRUE)
re <- CalEpletMHCII(dat, ver = 2)
```

CleanAllele *Clean messy HLA typing data*

Description

This function takes raw messy HLA(Human Leukocyte Antigen) typing data as input. It removes inconsistent formatting and unnecessary symbols. If one of two alleles at a loci is NA, the locus is assumed to be homozygous.

Usage

```
CleanAllele(var_1, var_2)
```

Arguments

var_1	HLA on allele 1.
var_2	HLA on allele 2.

Value

A data frame with 4 columns: - 'var_1': raw messy input hla, identical with first input - 'var_2': raw messy input hla, identical with second input - 'locus1_clean': cleaned hla of var_1 - 'locus2_clean': cleaned hla of var_2

Examples

```
dat <-  read.csv(system.file("extdata/example", "HLA_Clean_test.csv", package = "hlaR"))
re <- CleanAllele(dat$recipient_a1, dat$recipient_a2)
```

CountAlleleMism *Count HLA mismatch at the allele level*

Description

Donor and recipient HLA(Human Leukocyte Antigen) typing data is compared to determine allele level mismatch. The output of EvalAlleleMism is used as input for this function. Allele level mismatch can be calculated for both high and low resolution data. The generated count will return NA if the input alleles are NA.

Usage

```
CountAlleleMism(dat_in, names_in)
```

Arguments

dat_in	A data frame with donor and recipient mismatched alleles. It's a output from EvalAlleleMism function.
names_in	A vector of HLA loci name to count mismatch for.

Value

A tibble of input data (subject id and hla loci) followed by mismatch hla count of each subject.

Examples

```
hla <- read.csv(system.file("extdata/example", "HLA_MisMatch_count_test.csv", package = "hlaR"))
classI <- CountAlleleMism(hla, c("mism.a1", "mism.a2", "mism.b1", "mism.b2"))
classII <- CountAlleleMism(hla, c("mism.dqa12", "mism.dqb11", "mism.dqb12"))
```

EvalAlleleMism *Evaluate mismatched alleles*

Description

Compare donor and recipient HLA(Human Leukocyte Antigen) typing data to determine mismatched alleles. Input data can be high or low resolution, mismatch is evaluated at the allele level.

Usage

```
EvalAlleleMism(don_1, don_2, recip_1, recip_2, hmz_cnt = 1)
```

Arguments

don_1	Donor's alpha1 domain.
don_2	Donor's alpha2 or beta1 domain.
recip_1	Recipient's alpha1 domain.
recip_2	Recipient's alpha2 or beta1 domain.
hmz_cnt	Use hmz_cnt to determine how mismatch at homozygous alleles should be handled. By default, a mismatch at a homozygous allele is considered a single mismatch. Set hmz_cnt = 2 to count homozygous mismatches as double.

Value

A data frame of original input columns followed by mism_cnt of each donor/recipient pair.

Examples

```
dat <- read.csv(system.file("extdata/example", "HLA_Clean_test.csv", package = "hlaR"))
re <- EvalAlleleMism(dat$donor_a1, dat$donor_a2, dat$recipient_a1, dat$recipient_a2, hmz_cnt = 2)
```

ImputeHaplo *Imputation*

Description

Impute low or mixed resolution HLA(Human Leukocyte Antigen) typing to the most likely high resolution equivalent. Imputation is computationally intensive, so large dataset may encounter delays in processing. This function uses data from the NMDP(National Marrow Donor Program), and is currently limited to HLA A, B, C, and DRB loci.

Usage

```
ImputeHaplo(dat_in)
```

Arguments

dat_in	A data frame with low resolution HLA data.

Value

A data frame with high resolution HLA data pulled from the most likely pair of haplotypes matching the input low resolution data.

Examples

```
dat <- read.csv(system.file("extdata/example", "Haplotype_test.csv", package = "hlaR"))
result <- ImputeHaplo(dat_in = dat[c(1:2), ])
```

utils *Basic functions*

Description

GenerateLookup() called in CalEpletMHCII()

Usage

```
GenerateLookup(lkup_in, locus_in)

CalRiskScr(dat_scr)

FuncForCompHaplo(tbl_raw, tbl_in)
```

Arguments

lkup_in	data table
locus_in	string CalRiskScr() called in CalEpletMHCII()
dat_scr	dataframe FuncForCompHaplo() called in ImputeHaplo()
tbl_raw	data frame
tbl_in	data frame

Index